

There are a lot of product and project management applications that organizations can use to keep track of productivity- but none of them put a focus on the individual employees as much as they do the overall outcome of each project. This is why it is important to take a new approach to evaluating the performance of each individual on your team.

With this application managers can journal employee productivity and contribution every week, and a detailed evaluation will be generated for them at the end of each quarter. More advanced features are also included, such as tagging each individual member with strengths and weaknesses so that the system can generate teams based on what outcome you are looking for.

### ***Use Case:***

#### ***Automatic Team Formation***

A large advertising agency is contacted by a client that wishes to have a website built for them. The client wants the site to be professional with custom designs and a few advanced animations. Using this information, the agency uses my application's team composition feature and selects the tags 'professional design', 'html', and 'animation' as important. The application asks for the desired team size, and the manager selects 3. The application then chooses the 3 most capable employees to work on the project, and the team is formed.

#### ***Employee Chemistry***

The manager notices that during the project, two of the employees seem to have bad chemistry with one another. They can never seem to agree on the designs, and their bickering slowed down production remarkably. When evaluating these two individuals on the application, the manager adds one of the employees to the other's 'negative chemistry' list. Next time the team formation feature is used- it will take this into account and attempt to form a team that does not include both of them in it together.

#### ***Evaluations***

At the end of every week, the manager records any remarkable accomplishments for the team that he oversees. As you click on each employee, there are 5 options to choose from: Very Bad, Bad, Average, Good, and Great. The manager records his satisfaction with each of the employees and writes notes in the system when needed. At the end of the quarter, the application sends him a detailed report of each employee which includes the average satisfaction rating and every note that he has made throughout the quarter.

## Security:

*How do you make sure the system is secure?*

The system will be secured through user authentication, most likely through the use of JWT Tokens. Two factor authentication will also be available. Each user will only be able to see information based on their assigned roles. A team leader can have temporary access to employee evaluation- but they are not able to see previously taken notes. The main manager/administrator will have access to all notes and will always be able to create new notes. This hierarchical structure allows for data to remain secure and confidential.

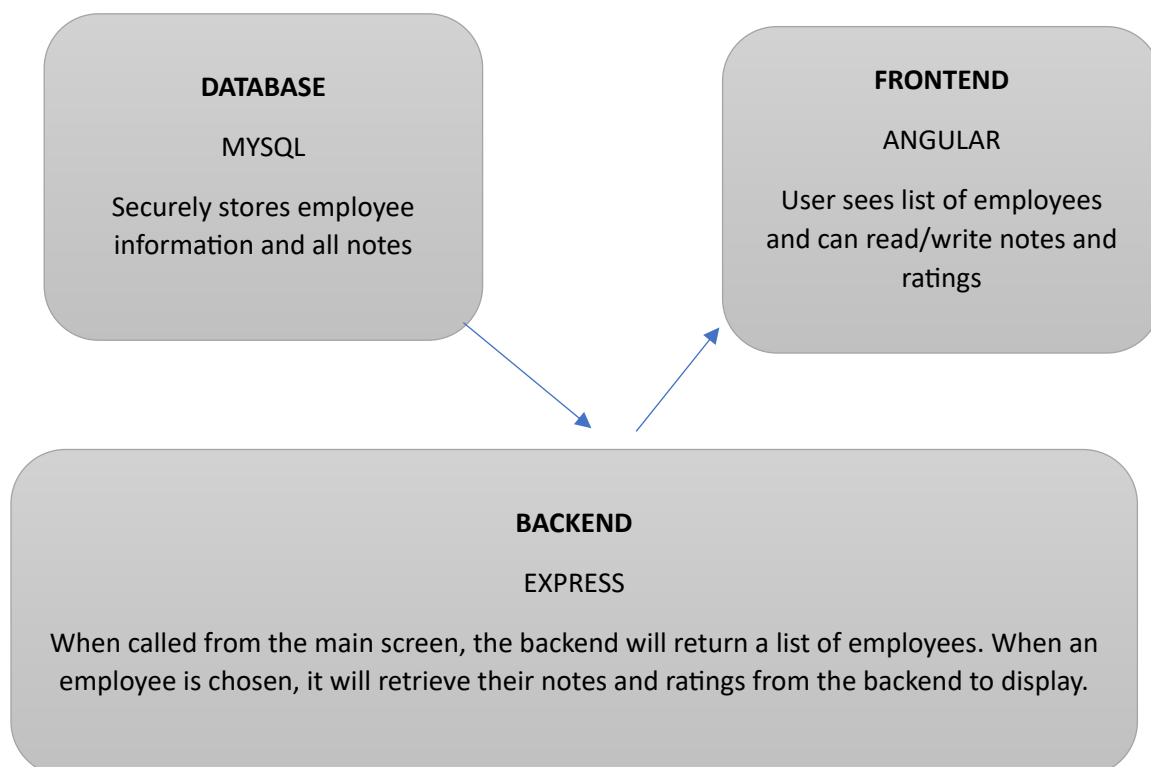
*What data is sensitive?*

Nearly all of the data stored in this application can be deemed confidential and personal. Each employee will have an array of notes and evaluations directly linked to their name, which immediately marks this data as personal.

*How will you protect it?*

Like with the system itself- security is of the utmost importance. The data related to each employee is personal and confidential. If somehow this data would get out or be visible to individuals of a lower-level clearance, employee relationships can be severely damaged. Even more so, if any negative or detrimental information leaked about an individual, their future employment opportunities may be compromised. Because of this, establishing a secure authentication process and making sure that there are no bugs related to role privileges is extremely critical to keep track of before launch.

## Software Architecture



## API Usage Example

User logs in to the splash page. The splash page must call the API to retrieve a list of employees in the organization.

\_\_\_\_.com/organization/team/

*Returns*

```
{
  'name1': {
    "name": xxx,
    "note_count": 7,
    "id": xxx,
    "notes": ["Good employee", "Did great this week"],
    "ratings": [5, 5, 4, 5, 5, 2, 5],
    "strengths": [xxx, xxx],
    "weaknesses": [xxx, xxx, xxx],
    "chemistry_positive": [xxx, xxx],
    "chemistry_negative": [xxx, xxx]
  }
  'name 2': {}
}
```

This makes it simple to retrieve individual data or to retrieve many pieces of information at once.